Using Artificial Intelligence Techniques in Computer Graphics

Dimitri PLEMENOS
MSI Laboratory, University of Limoges
plemenos@unilim.fr

Abstract:

The object of this paper is to prove that the use of artificial intelligence techniques in computer graphics can greatly improve the obtained results. Improvements in three areas of computer graphics are presented in the paper: in scene modelling, where declarative modelling techniques can greatly improve the designer's work ; in scene understanding, where the use of heuristic search and strategy games techniques allows the user to well understand a scene by means of a virtual camera moving around the scene; in Monte Carlo radiosity techniques where the use of heuristic search permits to automatically estimate the complexity of a region of the scene to be rendered and so to refine the processing of complex regions in order to obtain more accurate images.

Keywords: *Computer Graphics, Artificial Intelligence, Radiosity, Scene modelling, Scene understanding, Heuristic search, Rules-based systems, Constraint satisfaction, Strategy games.*

1. INTRODUCTION

Computer graphics is one of the three areas of graphics processing techniques, the two others being Image processing and Pattern recognition. In a general manner, computer graphics includes (fixed or animated) scene modelling and (fixed or animated) scene visualisation, this visualisation being photorealistic or not.

The two main phases of computer graphics, scene modelling and visualisation, have been first developed independently of each other. Scene visualisation techniques have been developed before scene modelling techniques in the years 1970, because the processing power of computers was then insufficient to process very complex scenes. So, it was not really a problem to create the simple scenes required by the rendering algorithms and adapted to the processing power of computers. During this period, very interesting visualisation algorithms have been developed.

When, at the end of years 70, computers became more powerful, people discovered that it was then possible to process complex scenes with the existing computers and algorithms but it was not possible to get complex scenes. Computer graphics researchers discovered that it is difficult to empirically design a scene by giving only coordinates of points or equations of curves or surfaces. Research on scene modelling has then begun and several models have been proposed in order to improve scene modelling. Currently, there exist well defined geometric models, used into powerful scene modellers, and the design of a scene is easier than ten or twenty years ago.

Even if today's scene modelling and rendering techniques are very powerful, there exist cases where the available tools are not entirely satisfactory, as well in scene modelling as in scene visualisation and rendering. In these cases, the use of Artificial Intelligence techniques can improve the modelling and rendering processes.

After having tried to explain why artificial intelligence is useful in computer graphics in section 2, applications of AI techniques in various areas of computer graphics will be studied. Thus, in section 3, applications of AI techniques in declarative modelling will be presented while in sections 4 and 5 using of AI techniques in, respectively, scene understanding and radiosity computation will be discussed. Section 6 will permit to conclude.

2. WHY ARTIFICIAL INTELLIGENCE IS USEFUL IN COMPUTER GRAPHICS

The two main areas of computer graphics, scene modelling and scene visualisation are currently well developed and allow to create and display rather complex scenes with a high degree of realism. However, several problems remain and artificial intelligence techniques could give a satisfactory solution to these problems. We are going here to give some examples of non resolved problems in computer graphics.

2.1 SCENE MODELLING

Geometric modelling is not well adapted to CAD. The main reasons of this are ([16, 17, 3]):

- the lack of abstraction levels in descriptions, which renders some information difficult to obtain,
- the impossibility to use approximative or unprecise descriptions to express unprecise mental images of the user. The user of a scene modelling system would like to express high level properties of a desired scene and would like to let the modeller construct all of the scenes verifying these properties.

2.2 SCENE UNDERSTANDING

Complex scenes are difficult to understand, especially scenes found on the web, because the scenes are three-dimensional and the screen two-dimensional and it is difficult to reach manually a good view position allowing to well understand a scene.

2.3 RADIOSITY

In Monte Carlo techniques used to compute radiosity, an important number of randomly chosen rays are shot from each patch of the scene. These rays permit to regularly sample the scene and to diffuse the energy of each patch. Unfortunately, regular sampling is not always well adapted because most of the scenes are not uniformly complex. Some regions of a scene can be complex enough while others can be very simple (e.g. a wall). How estimate the complexity of a region of a scene ?

2.4 OTHER PROBLEMS

Other important problems could obtain a satisfactory solution with artificial intelligence techniques: placement of a light source according to the expected results; design of scenes with specific properties, to be used to test new rendering methods, etc.

3. DECLARATIVE MODELLING

3.1 PRINCIPLE

The purpose of declarative modelling is to attenuate drawbacks of classical geometric modelling by offering the possibility of scene description using properties, which can be either precise or imprecise[3, 5, 13]).

More precisely, declarative modelling allows the user to tell which properties must verify a scene, without indicating the manner to obtain a scene with these properties. Because the scene's designer has not necessarily complete knowledge of all details of the scene he (she) wants to obtain, it seems natural to allow him (her) to use imprecise properties to describe the scene.

3.2 DECLARATIVE MODELLING USING MULTIFORMES

In order to allow descriptions at various detail levels, a new declarative modelling technique, so-called declarative modelling by hierarchical decomposition (DMHD) [3], has been introduced by the author. This modelling technique uses top-down hierarchical description and works as follows :

If a scene is easy to describe, it is described by a small number of properties which can actually be **size** (inter-dimensions) **properties** (higher than large, as high as deep, etc.) or **form properties** (elongated, very rounded, etc.). Otherwise the scene is partially described with properties easy to describe and is then decomposed into a number of sub-scenes and the same description process is applied to each sub-scene. In order to express relationships within sub-scenes of a scene, **placement properties** (put on, pasted on the left, etc.) and **size** (inter-scenes) **properties** (higher than, etc.) are used.

The DMHD technique has been put into practice in the MultiFormes modeller. This declarative modeller allows to describe scenes in hierarchical manner and it is able to generate all the possible scenes corresponding to the description. Scene generation permits to obtain scenes at various levels of detail and even to mix rough and detailed representations for the different parts of a scene.

3.3 THE AI TECHNIQUES USED

Three artificial intelligence techniques are used in MultiFormes: rules-based systems, constraint satisfaction methods and machine learning. Another artificial intelligence technique, heuristic search, is also used in MultiFormes but, as this technique is not specifically connected to declarative modelling, it will be evoked with other techniques presented in this paper, scene understanding and scene's complexity estimation in Monte Carlo radiosity.

The first version of MultiFormes used Prolog-like rules and an inference engine to generate all the solutions for a hierarchical scene's description. An external description of the scene given by the designer via an interface is converted into a set of Prolog-like rules. A special inference engine processes these rules and generates all the solutions corresponding to the description. The inference engine is able to take into account the hierarchical structure of the description and to generate scenes at various levels of detail [3, 5]. In figure 1, one can see an example of solutions generated by the first version of MultiFormes.

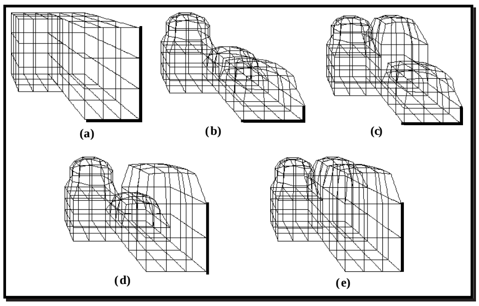

Figure 1: Scenes generated by an inference engine

This version of MultiFormes is not very efficient because each property is converted to a rule and, in this manner, a rule can be something very complex which is processed by the inference engine as a whole, with no possibility to be decomposed in order to improve its processing.

A second version of MultiFormes reduced the role of the inference engine and used linear constraints resolution to improve scene generation [5, 19]. This version is more efficient than the first one because each property is decomposed in a set of constraints and resolution improvements are applied to each constraint. However, the used constraints can be expressions of any length and this fact limits possibilities of improvement.

In the current version of MultiFormes, each property known by the modeller is described by a set of linear constraints. So, the property can be decomposed and special constraint satisfaction techniques can be used to improve the scene generation process. The constraint satisfaction technique used by MultiFormes consists to decompose each constraint describing a property in a set of simple arithmetic constraints and in a set of associated primitive constraints of the form "X in r", where X is a variable that takes its values in a finite domain and r is a range. This is the area of CLP(FD) (Constraint Logic Programming on Finite Domain) which was originally proposed by Pascal Van Hentenryck [22, 23]. In the case of MultiFormes, the resolution process is applied to primitive constraints, together with heuristics permitting to process the variables in an order defined by the hierarchical decomposition tree. This new technique permits to greatly improve the generation capability of MultiFormes. In figures 2 and 3, one can see two scenes generated with the last version of MultiFormes, a three-floors building and a chair respectively. Table 1 shows generation improvement of the last version of MultiFormes for the scene “chair”.

	Previous version of MultiFormes	Current version of MultiFormes
Number of backtracks	32284	7
Number of tries	74224	91
Time	2 min 27 sec	2 sec

Table 1: A comparison between the previous and the current version of MultiFormes for the scene “chair”

Machine learning mechanisms have also been associated to the scene generation engine of MultiFormes. These mechanisms are based on a neural network, dynamically generated from the scene’s description, and permit to generate only solutions that could be interesting for the designer [14, 15, 18].

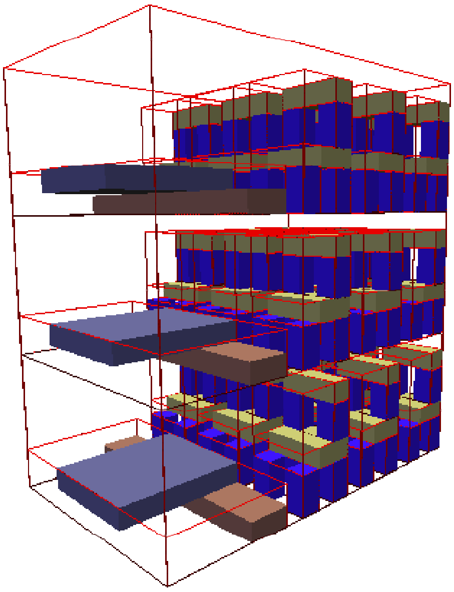

Figure 2: A three-floors building

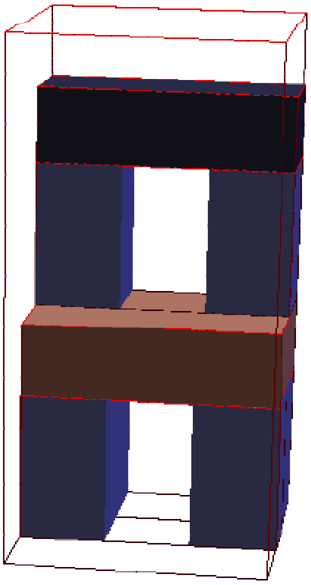

Figure 3: A chair

4. SCENE UNDERSTANDING

In order to understand a scene, designed with a modeller or found in the net, it is important to choose a view direction which shows its most important features. Such a view direction is very difficult to find interactively because the scene is generally 3-dimensional while the screen is 2-dimensional. Thus, it is very important that a scene modeller offers an automated computing of a good view direction. Indeed, the modeller has much more information about the scene than the user and could use this information to automaticaly compute a good view direction. This is especially true for declarative modeling, where the designer has insufficient knowledge of the scene during the designing process.

4.1 STATIC UNDERSTANDING

In order to allow automatic computing of a good view direction, we have developed a method [3, 4, 6] using a heuristic search. From a criterion of *good view* it applies a heuristic based on the evaluation of some view directions from which it computes other directions assumed better. These new directions are inferred from the hypothesis that a direction near a good direction is also probably a good direction. To do this, the method uses a sphere surrounding the scene (figure 4) and good points of view are computed in the surface of the sphere by subdividing the sphere in spherical triangles.

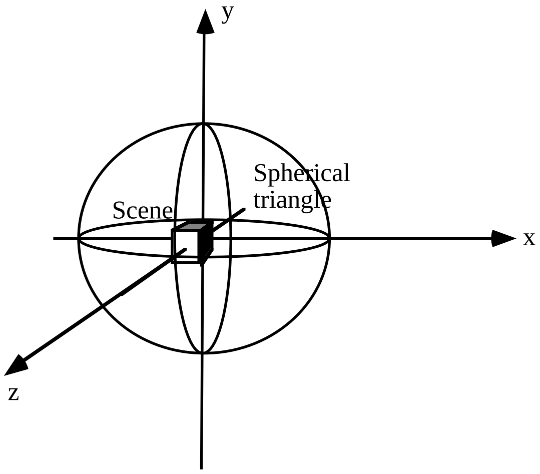

Figure 4: Sphere surrounding a scene and spherical triangles

This method was improved by introducing an additional “good view” criterion, based on the area of the projected total visible part of the scene.This new criterion is combined with the criterion of number of visible surfaces. An example of obtained results with a simple scene is shown in figure 5.

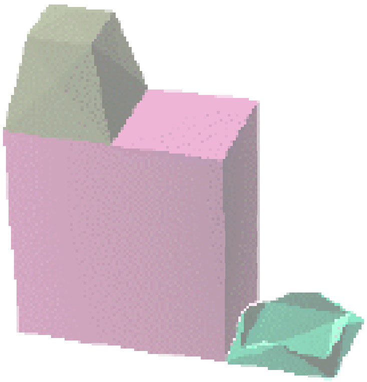

Figure 5: Good view direction for a simple scene

Other methods for computing good points of view have been proposed, especially by Colin and Kamada [1, 2].

4.2 DYNAMIC UNDERSTANDING

The computation of a single good direction is not sufficient, in many cases, to have a good knowledge of a scene. For some scenes, several views are necessary to well understand their properties. The work of computing these views can be left to the modeller. The problem is that changing a view direction for another one can be confusing for the user, especially if the new view direction is completely different from the previous one.

A way to avoid brutal changes of view direction is to simulate a virtual camera moving smoothly around the scene. In this camera’s movement, sudden changes of the camera’s path must be avoided in order to have a smooth movement of the camera, and heuristics must be provided to avoid attraction forces in the neighbouring of a good view direction.

The camera moves on the surface of a sphere surrounding the scene (figure 6). The starting point of the camera is computed using the method of calculation of a good view direction described above and in [6].

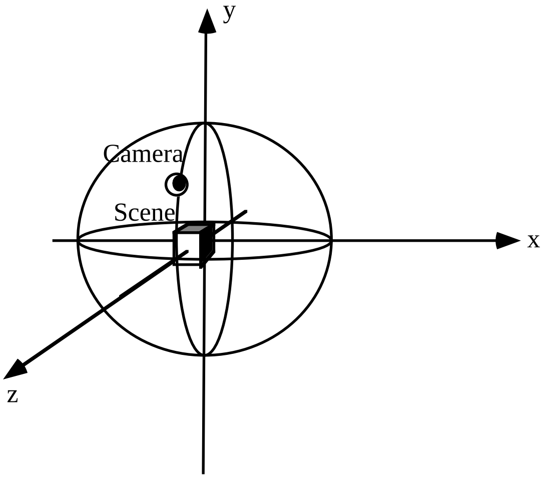

Figure 6: The virtual camera moves on the surface of the sphere

After the first displacement of the camera, a movement direction is defined by the previous and the current position of the camera. As blunt changes of movement direction have to be avoided, in order to obtain a smooth movement of the camera, the number of possible new directions of the camera is reduced and only 3 directions are possible for each new displacement of the camera (figure 7).

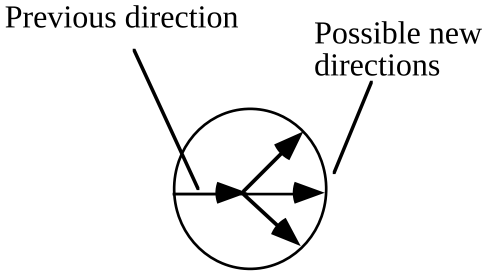

Figure 7: Only 3 directions are considered for a smooth movement of the camera

One of the three possible directions is chosen using heuristic rules taking into account not only the view direction value of a point but also other parameters permitting to avoid attraction points and cycles in the camera’s movement.

As the importance of the camera’s distance from the starting point is inversely proportional to the length of the path traced by the camera, the basic heuristic function computing the weight of a position for the camera on the surface of the sphere takes into account :

- The global view direction note of the camera’s position (n_c).
- The path traced by the camera from the starting point to the current position (p_c).
- The distance of the current position from the starting point (d_c).

Thus, the basic function computing the movement of the camera is :

$$w_c = \frac{n_c}{2} \left(1+\frac{d_c}{p_c}\right)$$

where w denotes the weight and c the current position of the camera. Other heuristics have been added in the basic function in order to improve exploration of the scene by the camera [10, 11, 12]. One of these heuristics is the use of evaluation at various levels of depth, as in many strategy games. The obtained results are very satisfactory. Figure 9 shows the path of a virtual camera exploring a scene representing an office.

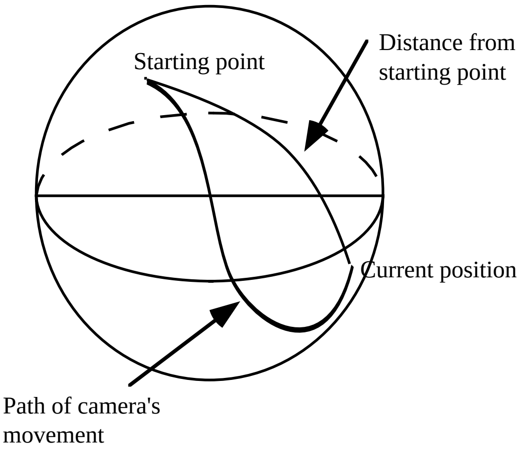

Figure 8: Path of the virtual camera and distance from its starting point

Some authors have also worked in the area of virtual camera's movement in relation with declarative modelling [7, 8, 9].

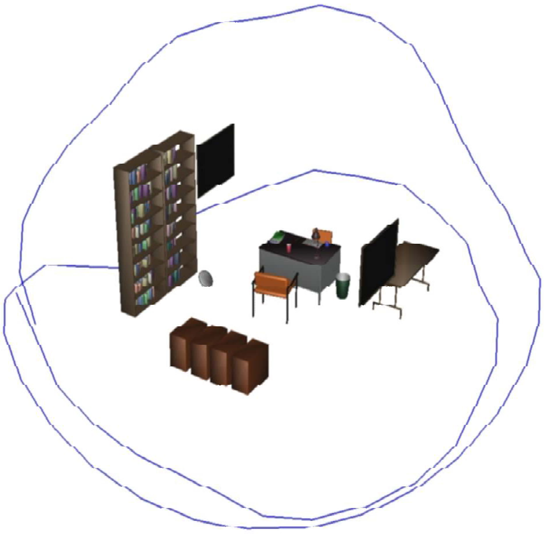

Figure 9: Exploration of a scene by a virtual camera

A new technique, using also heuristic search, has been developed in order to permit the camera to move inside the scene and so to improve scene's understanding [12]. In figure 10, the path of the virtual camera inside a seashell is shown.

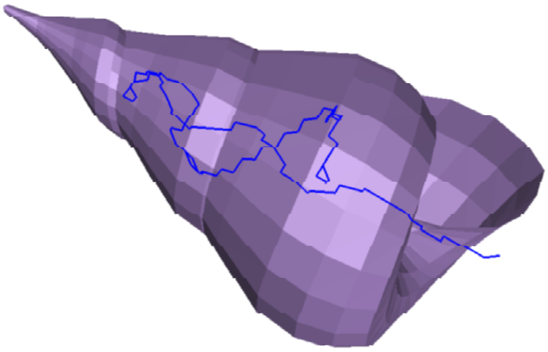

Figure 10: Path of a virtual camera inside a seashell

5. SCENE RENDERING

5.1 IMPROVING MONTE CARLO RADIOSITY TECHNIQUES

Techniques for improve the precision of Monte Carlo radiosity have been developed since 1996 [4, 24, 25], These techniques are based upon hemisphere subdivision and their main purpose is to estimate the visual complexity of a scene from a patch and to use this complexity in order to :

- Get more precision in radiosity computation with Monte Carlo based algorithms, by shooting more rays in directions where the scene is more complex.
- Get a useful image more quickly, permitting to understand the scene and to modify it if visual impression is not satisfactory.

Heuristic search techniques are used to estimate the visual complexity of the regions of a scene. The results of these techniques are more spectacular for scenes containing altogether simple and complex parts.

5.2 SAMPLING CRITERIA

In order to make the estimation of the visual complexity of a region of the scene precise enough, the notion of density of a region have been introduced [24]. The density of a region viewed from a patch is the number of objects (patches) contained in the region. A hemisphere, divided in four spherical triangles, is associated to each processed patch of the scene. At the beginning, the hemisphere is divided into 4 equal-sized spherical triangles (figure 11). Each spherical triangle is then subdivided independently of the others, according to the retained criterion.

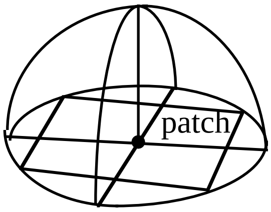

Figure 11: Initial subdivision of a hemisphere

As regions are delimited by spherical triangles in our case, the density of a region viewed from a patch is the number of objects (patches) contained in the triangular pyramid defined by the centre of the patch and the three vertices of the spherical triangle (figure 12).

This global density of a triangular pyramid is used in the following manner for computing radiosity, depending on the use of regular or adaptive subdivision.

Regular subdivision of the spherical triangles of the hemisphere corresponding to each patch is performed up to a subdivision level defined at the beginning of the process.

At the end of the subdivision phase, a number of rays, transporting the energy of the patch, are shot in the hemisphere. The number of rays shot in a region delimited by a spherical triangle is proportional to its global density. The amount of energy transported by a ray is proportional to the area of the corresponding spherical triangle.

Adaptive subdivision of the spherical triangles of the hemisphere corresponding to each patch is performed up to a subdivision level defined at the beginning of the process, only for triangles corresponding to pyramids with global densities greater than a threshold value.

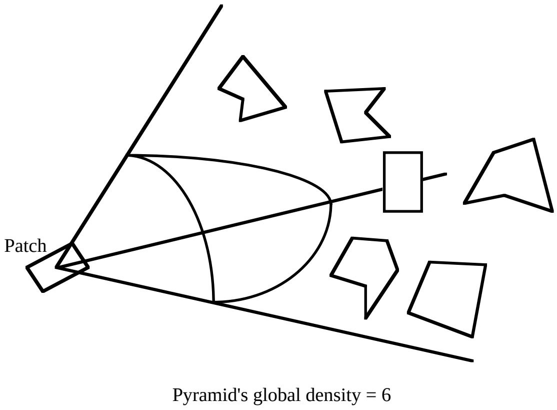

Figure 12: Global density of a pyramidal region

The energy diffusion phase for the current patch is performed at the end of the subdivision phase by shooting a number of rays in the hemisphere. The number of rays shot in a spherical triangle is the same for all spherical triangles. The amount of energy transported by a ray is proportional to the area of the corresponding spherical triangle.

Figure 13 shows a rather complex detail of a scene rendered by traditional Monte Carlo radiosity while figure 14 shows the same detail of the reference scene rendered by the pyramidal hemisphere subdivision technique.

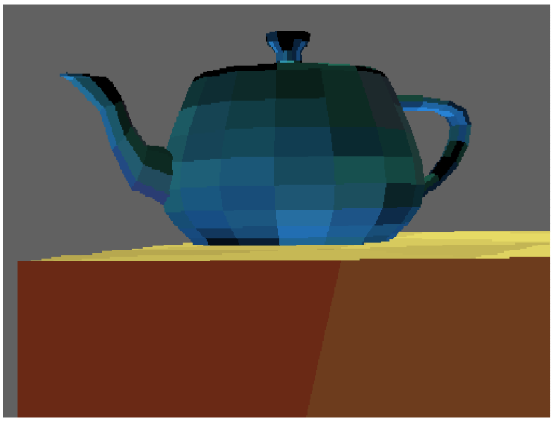

Figure 13: Detail of the reference scene obtained by traditional Monte Carlo radiosity after 1 step

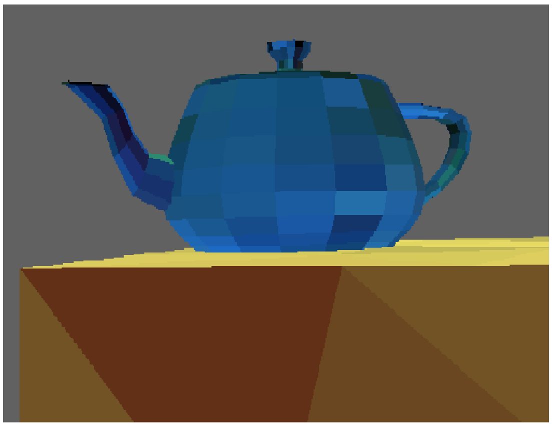

Figure 14: Detail of the reference scene obtained by pyramidal hemisphere subdivision after 1 step

6. CONCLUSION

In this paper we have presented some possibilities of using Artificial Intelligence techniques in various areas of Computer Graphics. The obtained results show that, in many cases, the use of Artificial Intelligence techniques can improve drastically the modelling and rendering processes. The main artificial intelligence techniques used are rules-based systems, constraint satisfaction methods and machine learning, in scene modelling; heuristic search and strategy games techniques in scene understanding and scene rendering.

Currently we are working on some other problems whose solution should be improved by the use of artificial intelligence techniques. These problems are: modelling of non geometric properties of a scene, placement of a light source according to the expected results and design of scenes with specific properties, to be used for testing new rendering methods.

7. REFERENCES

[1] COLIN C., Automatic computing of good views of a scene,

MICAD'90, Paris (France), February 1990 (in French).

[2] KAMADA T., KAWAI S., A Simple Method for Computing General Position in Displaying Three-dimensional Objects, Computer Vision, Graphics and Image Processing, 41 (1988).

[3] PLEMENOS D., Contribution to the study and development of techniques of modelling, generation and display of scenes. The MultiFormes project., Professorial dissertation, Nantes (France), November 1991 (in French).

[4] PLEMENOS D., PUEYO X., Heuristic Sampling Techniques for Shooting Rays in Radiosity Algorithms., 3IA'96 International Conference, Limoges (France), April 3-4, 1996.

[5] PLEMENOS D., Declarative modeling by hierarchical decomposition. The actual state of the MultiFormes project., GraphiCon'95, Saint Petersburg (Russia), July 3-7, 1995.

[6] PLEMENOS D., BENAYADA M., Intelligent display in scene modelling. New techniques to automatically compute good views., GraphiCon'96, Saint Petersburg, July 1996.

[7] FOURNIER L., Automotive vehicle and environment modeling. Machine learning algorithms for automatic transmission control. PhD Thesis, Limoges (France), January 1996 (in French).

[8] MOUNIER J.-P., The use of genetic algorithms for path planning in the field of declarative modeling., International Conference 3IA'98, Limoges (FRANCE), April 28-29 1998.

[9] F. JARDILLIER, E. LANGUENOU, Screen-Space Constraints for Camera Movements: the Virtual Cameraman., Computer Graphics Forum, Volume 17, number 3, 1998.

[10] BARRAL P., DORME G., PLEMENOS D., Visual understanding of a scene by automatic movement of a camera, GraphiCon'99, Moscow (Russia), August 26 - September 1, 1999.

[11] BARRAL P., DORME G., PLEMENOS D., Visual understanding of a scene by automatic movement of a camera, Eurographics 2000, short paper.

[12] BARRAL P., DORME G., PLEMENOS D., Visual understanding of a scene by automatic movement of a camera, International conference 3IA'2000, Limoges (France), March 3-4, 2000.

[13] LUCAS M., MARTIN D., MARTIN P., PLEMENOS D., The ExploFormes project : some steps towards declarative modeling of forms (in french). Journées AFCET-GROPLAN, Strasbourg (France), 29 november to 1st december 1989. Published in BIGRE, no 67, january 1990, pp 35 - 49.

[14] Mc CULLOCH W. S., PITTS W., A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5, 115 - 133, 1943.

[15] ROSENBLATT F., The perceptron : a perceiving and recognizing automaton. Project Para, Cornell Aeronautical Lab. Report 85-460-1, 1957.

[16] SHAH J. I., ROGERS M. T.
Expert form feature modelling shell. CAD, vol. 20, no 9, november 1988.

[17] SMITHERS T.
AI-based design versus geometry based design ot why design cannot be supported by geometry alone. CAD, vol. 21, no 3, april 1989.

[18] BOUGHANEM M., PLEMENOS D. Learning techniques in declarative modeling of scenes by hierarchical decomposition (in french). Conference 3IA'94, Limoges (France), april 1994.

[19] PLEMENOS D., TAMINE K., Increasing the efficiency of declarative modelling. Constraint evaluation for the hierarchical decomposition approach. International Conference WSCG'97, Plzen (Czech Republic), February 1997.

[20] BONNEFOI P.-F., Constraint satisfaction techniques for declarative modelling (in French). PhD Thesis, June 25, 1999, Limoges (France).

[21] BONNEFOI P.-F., PLEMENOS D., Constraint satisfaction techniques for declarative scene modelling by hierarchical decomposition. International Conference 3IA'2000, May 3-4, 2000, Limoges (France).

[22] VAN HENTENRYCK P., Constraint satisfaction in logic programming. Logic Programming Series, MIT Press, 1989.

[23] DIAZ D., A study of compiling techniques for logic languages, for programming by constraints on finite domains: the clp(FD) system. PhD thesis, Orléans (France), 1995 (in French).

[24] JOLIVET V., PLEMENOS D., SBERT M., A pyramidal hemisphere subdivision method for Monte Carlo radiosity, Eurographics'99 (short paper), Milano (Italy), September 1999.

[25] JOLIVET V., PLEMENOS D., SBERT M., Pyramidal hemisphere subdivision radiosity. Definition and improvements, proceedings of the WSCG'2000 International Conference, Plzen (Czech Republic), February 2000.

Author:

Dimitri PLEMENOS is full professor at the University of Limoges. His research area is computer graphics and, more precisely, the use of artificial intelligence techniques in computer graphics. He is the director of the MSI research laboratory.

Dimitri PLEMENOS
University of Limoges
MSI Laboratory
83, rue d'Isle
87000 Limoges
France
Phone: (+33) 555 43 69 74
Fax: (+33) 555 43 69 77
E-mail: plemenos@unilim.fr